REMARKS

Reconsideration and allowance of the above-referenced application are respectfully requested. Claims 1 and 8 are amended, and claims 1-13 are pending in the application. Support for the amendments are found, for example, at page 10, lines 21-25 and 29-34 of the specification, and steps 104 and 106 of Fig. 3A.

Claims 1-12 were stand rejected under 35 USC §103 in view of U.S. Patent No. 6,243,778 to Fung and U.S. Patent No. 6,199,137 to Aguilar. This rejection is respectfully traversed.

The claims specify an arrangement that reduces the writing of entries in a retransmission table to an end of each access cycle, enabling the access cycle to be reduced. Claim 1 is exemplary:

> 1. A method in a host channel adapter, the method comprising:
>
> first storing in a table, at an end of each access cycle by a retransmission manager, **entries identifying respective packets**, said packets having been transmitted onto an InfiniBand™ network according to InfiniBand™ protocol and during said each access cycle according to *an InfiniBand™-based service protocol requiring an acknowledgement message receipt within a prescribed time interval following transmission of the corresponding packet*;
>
> resetting an acknowledgment waiting bit for a selected one of the entries by an acknowledgement manager in response to reception of an acknowledgment message identifying the corresponding packet identified by the selected one entry; and
>
> transferring the entries having a determined absence of the reset stored acknowledgment waiting bit *upon expiration of the prescribed time interval* to a transmit queue for retransmission onto the InfiniBand™ network according to InfiniBand™ protocol.

As illustrated on page 2, line 23 of the specification, the "access cycle" is "defined as a prescribed number of clock cycles." In addition, the specification describes that the retransmission manger stores the entries at the end of each access cycle (i.e., after every "n" clock cycles) to enable the memory size to be reduced.

> The retransmission manager 24 is configured for storing the entries 26 into the retransmission table 20 during each access cycle (i.e., after every "n" clock cycles). As described in detail below with respect to Figures 3A and 3B, the retransmission manager 24 writes the entries for the packet transmitted during the access cycle by accessing the

> retransmission table after every "n" clock cycles, reducing the number of access attempts to the retransmission table 20. In particular, the restriction of writing the entries at prescribed access cycles *enables the memory size to be reduced*, since any attempt to write an entry into the table 20 each clock cycle would result in a large memory size if the retransmission time "t" was a large number. *Hence, the size of the table 20 can be reduced based on limiting the access to a prescribed number of clock cycles for each access attempt.*
>
> (Page 10, lines 17-25).

The specification also describes that the retransmission manager monitors the packets transmitted during the access cycle, and adds the entries into the retransmission table at the end of the access cycle:

> The method begins in step 100, wherein the access cycle starts by the retransmission manager 24 monitoring in step 102 the packets transmitted by the MAC 74 onto the network, and counting the number of transmitted packets using the counter 22. The retransmission manager 24 continues monitoring the packets having been transmitted until determining in step 104 that a prescribed number (n) of clock cycles have passed, indicating the access cycle is complete. Once the prescribed number of clock cycles have passed, the retransmission manager 24 stores in step 106 the entries 26 into the retransmission table 20 identifying the transmitted packets, based on the queue pair number field 28 and the packet sequence number field 30.
>
> (Page 10, lines 27-34).

In addition, the claimed "prescribed time interval" is distinct from the claimed access cycle, since the claimed "prescribed time interval" is based on the claimed *service protocol.* Claim 1 specifies "said packets having been transmitted ... according to an InfiniBand™-based service protocol requiring an acknowledgement message receipt within a prescribed time interval following transmission of the corresponding packet". In addition, the entries that have a determined absence of the reset stored acknowledgement waiting bit upon expiration of the prescribed time interval are transferred to the transmit queue.

Hence, the retransmission manager stores entries at the end of each access cycle (described in the specification as upon expiration of "n" clock cycles), enabling the size of the table to be reduced. Further, the storage of entries identifying respective packets, as opposed to a complete transaction, enables the retransmission of only those packets not having a reset stored acknowledgement waiting bit (see, e.g., page 12, lines 1-8 and step 128 of Fig. 3B) after a prescribed time interval, as opposed to complete retransmission of all packets having been transmitted during the given access cycle.

Hence, the claimed storage of entries identifying respective packets minimizes the packets that need to be retransmitted for a given access cycle. Moreover, the storage of entries at an end of each access cycle limits the access attempts for storing the entries in the table, enabling the table size to be reduced. These and other features are neither disclosed nor suggested in the applied prior art.

Applicant hereby incorporates by reference the prior arguments submitted June 14, 2004 and January 30, 2004 related to Fung et al. The Examiner to date still has failed to identify any description whatsoever in Fung et al. that teaches or suggest the claimed storage of entries identifying respective packets. Rather, as described on pages 6-7 of the June 14, 2004 response, Fung et al. stores TMC blocks on a per-transaction basis, where each transaction may include a plurality of data packets. For example, Fung et al. specifies:

> The transaction interface includes a queue that accepts message control blocks, which contain organized data, a conversion engine that reads the message control blocks and converts them into data packets ....
>
> (Col. 3, lines 4-8).
>
> If the amount of data to be sent from the Transaction Interface 210 is large, it may be broken up into several packets to be placed on the bus. Each packet is prepared and then sent along the bus.
>
> (Col. 10, lines 47-50).

Further, as described in column 11, the Transaction Interface 210 creates transaction memory control (TMC) block 310 for every transaction. However, the TMC block as illustrated in Figure 5A has a "transaction_count" to keep track of the number of pending responses relative to the number of transmitted packets associated with the transaction:

> The "transaction_count" in bits 0-3 of word 0 enables the Transaction Interface 210 to track the number of outstanding transaction requests. Each time the Transaction Interface 210 sends a data packet to the serial bus, it increments transaction_count. Every time the Transaction Interface 210 receives a transaction response from the node to which the data packet was sent, the Transaction Interface decrements the transaction_count. In this way, the Transaction Interface 210 always knows *how many outstanding transaction responses for which it is waiting.*
>
> (Col. 11, lines 41-50).

> The "transaction_count" entry records the number of outstanding transaction requests for the particular TMC block. For instance, if a requesting task is sending ten blocks to another node, after each block is sent by the Transaction Hardware 205, the "transaction_count" is incremented by one. Every time a transaction response is received for that TMC block, the "transaction_count" is decremented by one. Therefore, if the "transaction_count" field within the TMC block 310 is anything other than zero, packets have been sent to a node on the 1394 bus, but no response received.
>
> (Col. 17, line 63 to col. 18, line 6).

Contrary to the assertions of the Examiner on pages 5-7 of the Office Action mailed April 13, 2004, Fung et al. does not disclose retransmission of entries identifying respective packets: rather, Fung et al. describes retransmission of the entire transaction that is composed of multiple packets:

> Each transaction that is initiated by the Transaction Interface 210 has a hardware timer associate [sic] with it. The hardware timer is used for timekeeping the transaction timeout. A retry count field of the TMC block is incremented if the data transmission is unsuccessful. As long as the retry count is below the programmable maximum number of retries, the Transaction Interface 210 will attempt to send *the data* again.
>
> (Col. 10, line 66 to col. 11, line 5).

As apparent from the foregoing, Fung et al. teaches retransmitting the entire transaction, which would include packets for which a reply already has been received. As demonstrated above, the assertion by the Examiner on page 6 of the April 13, 2004 that "Fung's transaction interface stores *an entry* [sic] to identify respective packets. This is clear from the cited passage which teaches associating a hardware timer with *each transaction*" is imprecise and fails to address the explicit claim limitations that require each and every packet to have a corresponding entry.

Hence, the Examiner's interpretation of the claimed "packet" as broad enough to encompass the "transaction" of Fung is inconsistent with both the specification, as well as the explicit teachings of Fung. It is notoriously well known that "claims are not to be read in a vacuum, and limitations therein are to be interpreted in light of the specification in giving them their 'broadest reasonable interpretation.'" MPEP § 2111.01 at 2100-37 (Rev. 1, Feb. 2000) (quoting In re Marosi, 218 USPQ 289, 292 (Fed. Cir. 1983)(emphasis in original)). A prior art reference must be considered in its entirety, i.e., as a whole, including portions that would lead away from the claimed invention. MPEP §2141.02, page 2100-127 (Rev. 2, May 2004) (citing W.L. Gore & Assoc. v. Garlock, Inc., 220 USPQ 303 (Fed. Cir. 1983), cert. denied, 469 U.S. 851 (1984)).

The addition of Alvaro to the hypothetical combination provides no additional teaching that is relevant to the arguments presented above. Hence, the Official Action fails to establish a prima facie case of obviousness. Hence, this rejection should be withdrawn.

In view of the above, it is believed this application is in condition for allowance, and such a Notice is respectfully solicited.

To the extent necessary, Applicant petitions for an extension of time under 37 C.F.R. 1.136. Please charge any shortage in fees due in connection with the filing of this paper, including any missing or insufficient fees under 37 C.F.R. 1.17(a), to Deposit Account No. 50-0687, under Order No. 95-391, and please credit any excess fees to such deposit account.

Respectfully submitted,

Manelli Denison & Selter, PLLC

Leon R. Turkevich
Registration No. 34,035

Customer No. 20736
Date: February 1, 2005